What is claimed is:

1. An execution unit for use in a computer system for conditionally carrying out an operation defined in a computer instruction, the execution unit comprising:

first and second input stores for holding respective first and second operands on which an operation defined in the instruction is to be carried out, wherein each store defines a plurality of lanes each holding an object;

a plurality of operators associated respectively with the lanes for carrying out an operation specified in the instruction on objects in corresponding lanes of the first and second source operands;

a destination buffer for holding the results of the operation on a lane-by-lane basis; and

selecting means for determining for each lane in dependence on stored condition values whether or not the operation is to be executed on objects in that lane.

2. An execution unit according to claim 1, wherein said condition values comprise a set of condition codes.

3. An execution unit according to claim 2, wherein the selecting means comprises means for comparing selected ones of said set of condition codes with a test code identified in the instruction.

4. An execution unit according to claim 2, wherein the number of condition codes in said set corresponds to the maximum number of lanes in the first and second source operands.

5. An execution unit according to claim 2, which comprises a condition code generator for generating said set of condition codes responsive to execution of an instruction.

6. An execution unit according to claim 2, wherein the number of condition codes

in said set corresponds to the maximum number of lanes in the first and second source operands.

7. An execution unit according to claim 4, wherein the number of condition codes in said set corresponds to the maximum number of lanes in the first and second source operands and wherein the condition code generator is operable to generate the set of condition codes so that, when the operands have less than the maximum number of lanes, two or more condition codes are set to the same value so that each individual condition code in the set is generated regardless of the degree of packing of the first and second source operands.

8. A computer system for conditionally carrying out an operation defined in a computer instruction, the computer system comprising:

fetch and decode circuitry for fetching and decoding a sequence of instructions from a program memory;

at least one execution unit according to any of claims 1 to 7; and

at least one memory access unit for effecting memory access operations responsive to memory access instructions.

9. A computer system according to claim 8, which comprises a condition code register for holding said condition values in the form of a set of condition codes.

10. A computer system according to claim 8, which includes a test register for holding a test code, the test register being addressed by a computer instruction and said test code being used in comparison with said condition values to determine for each lane whether or not the operation is to be executed on objects in that lane.

11. A method of executing instructions on operands containing a plurality of packed objects, the method comprising:

accessing at least one source operand containing a plurality of packed objects in respective lanes;

accessing stored condition values to determine for each lane whether or not

an operation defined in the instruction is to be implemented on that lane of the operand; and

carrying out the operation and updating a destination operand only in those lanes for which the stored condition value indicates that the operation should be implemented.

11. A method according to claim 10, wherein the step of accessing the stored condition values comprises accessing a set of condition codes held in a condition code register and comparing said selected codes with a test code identified in the instruction.

12. A method according to claim 11, wherein the test code is held in a test register which is identified by an address in the instruction.